Final Report on EOARD Project

Comparing Human Concept Acquisition to Models in a Cognitive Architecture

Richard M Young and Anna L Cox
Psychology Department, University of Hertfordshire, UK
{r.m.young,a.cox}@herts.ac.uk

August 2002

Project no. SPC 014040
Reporting period: 1 October 2001 - 30 June 2002

Summary

A study funded by UK DERA at the University of Nottingham in the mid-1990s examined performance on a concept formation task, where subjects had to classify schematic aeroplanes as being either 'USA' or 'Australian'. Subjects displayed poor performance, but more intriguingly exhibited a wide range of variability. A simple model of the same task, constructed in ACT-R, also displayed great variability from run to run.

The present project aimed to investigate the reasons for the variability in the model, and if possible also in human subjects; and also to understand better the nature of concept representation in this class of model.

This report, (a) explains the model, (b) comments critically on aspects of the original studies, (c) analyses components of the variability, (d) offers an account for the variability in terms of random walk processes within the ACT-R learning mechanism, and (e) outlines a graphical depiction of the representation and gradual acquisition of the concept within the model.

Acknowledgements. We are grateful to our colleague Diana Kornbrot for discussions and for help especially with the statistical analysis reported in Section 3. Frank Ritter and Gordon Baxter kindly made available to us some time ago both the data and the Soar model from the DERA-funded study at Nottingham University. Martin Greaves captured the information on model runs for the random walk analysis reported in Section 5.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YYYY)***	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
22-08-2002	Final Report	1 October 2001 - 01-Jul-02

4. TITLE AND SUBTITLE
Comparing Human Concept Acquisition to Models in a Cognitive Architecture

5a. CONTRACT NUMBER
F61775-01-WE040

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Richard Young

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Hertfordshire
Hatfield Herts AL10 9AB
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
EOARD
PSC 802 BOX 14
FPO 09499-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
SPC 01-4040

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report results from a contract tasking University of Hertfordshire as follows: The contractor will investigate the comparison between human subjects on a defined concept acquisition task (based on airplane identification) and models within a cognitive architecture, focusing on the ACT-R model. Specifically, the contractor will investigate whether the model or close variants can account for the performance and variability of subjects, relating this to broader questions of subjects' behavior and classes of cognitive modeling.

15. SUBJECT TERMS
EOARD, Human Factors, Psychology, Cognition

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UL	32	Valerie Martindale, Maj, USAF
UNCLAS	UNCLAS	UNCLAS			**19b. TELEPHONE NUMBER** *(Include area code)* +44 (0)20 7514 4437

Contract F61775-01-WE040
Title: COMPARING HUMAN CONCEPT ACQUISITION TO MODELS IN A COGNITIVE ARCHITECTURE

This material is based upon work supported by the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory, under Contract No. F61775-01-WE040.

"The Contractor, University of Hertfordshire, hereby delcares that, to the best of its knowledge and belief, the technical data delivered herewith under Contract No. F61775-01-WE040 is complete, accurate, and complies with all requirements of the contract.

DATE: 25 Sep 2002

Name and Title of Authorized Official: Prof R.M. Young

I certify that there were no subject inventions to declare as defined in FAR 52.227-13, during the performance of this contract.

DATE: 25 SEP 2002

Name and Title of Authorized Official: Prof. R. M. Young

Final Report on EOARD Project

Comparing Human Concept Acquisition to Models in a Cognitive Architecture

Richard M Young and Anna L Cox
Psychology Department, University of Hertfordshire, UK

August 2002

1. Introduction

In the mid-1990s, the UK Defence Evaluation and Research Agency (DERA) funded a study at Nottingham University investigating human concept acquisition and its modelling within a cognitive architecture. There were two outputs from the project, though both are unpublished. One is a short, draft (and somewhat cryptic) report by McNaught & Gilmore (1996) on the results of their experiments, and the other is a program written by Baxter (1997) in the Soar cognitive architecture (Newell, 1990) for performing the Nottingham concept learning tasks, based upon the SCA model ("Symbolic Concept Acquisition") of Miller & Laird (1996). The two sides of the Nottingham project — empirical and modelling — do not seem to have been brought together within the scope of the DERA-funded study. A later undergraduate project at the University of Hertfordshire (Oni, 1998), supervised by one of the present authors (RMY), took some initial steps towards comparing the predictions of the model with the empirical data. Oni's results showed that with a little tuning, the Soar model could be made to match approximately the overall average performance of the experimental subjects, but there were some anomalies in the fits (not to mention apparent anomalies in the data: see Section 4 below), and it was unclear how to move forward with the model.

A striking outcome from Oni's (1998) examination of the Nottingham data was the extreme variability among subjects' data. The error rates are spread over a surprisingly broad range, with some subjects apparently performing consistently worse than chance on a 2-way classification.

Later, as an exercise at an ACT-R Summer School, Young (1999) built a simple model — arguably *the* simplest model possible in ACT-R — of performance on the Nottingham concept learning task. Young's focus at the time was on the very long-term dynamical behaviour of the model — for example training the model on tens of thousands of trials where it could no longer possibly correspond to the behaviour of real human subjects — and on the question of whether the model would spontaneously learn to encode only the relevant perceptual attributes. However, there was also some speculation as to whether the quasi-chaotic dynamical behaviour might throw light on the observed variability among the Nottingham subjects.

The present study attempts to bring together these various disparate strands. Using the Nottingham concept formation task (McNaught & Gilmore, 1996) and Young's (1999) ACT-R model, the study had two primary aims:

- to investigate the cause of the observed variability in runs of the model, and — by implication — in the subjects' performance;
- to understand better the nature of the concept representation in models of the class exemplified by Young's (1999) model.

The structure of the report is as follows. The next section summarises the relevant background, including the original experimental study by Allen & Brooks (1991), the Nottingham study devised as a replication but with schematic aeroplanes as the concept material (McNaught & Gilmore, 1996), and the structure of the model investigated here (Young, 1999). Section 3 reports on the fit between the model and the data, and on a preliminary analysis of the variability in performance. Section 4 comments critically on aspects of the Allen & Brooks and McNaught & Gilmore studies. Section 5 offers an account of the variability in terms of random walk processes within the ACT-R learning mechanism, and Section 6 sketches a graphical depiction of the gradual acquisition of the concept within the model. An Appendix contains the model code. We also began exploring some ideas about the methodology of fitting process models to data by estimating best-fit parameter values, but the ideas are still too underdeveloped to report.

2. Background

As just summarised, the present study builds upon various pieces of previous work, relevant aspects of which are described in this section.

2.1 The Allen & Brooks study

Allen & Brooks' (1991) present their concept learning experiment in the context of a theoretical question about the nature of learned concepts. They believe that even long after a classification rule has been learned and frequently applied, specific episodes — taken to mean the original training of the classification of individual stimuli — continue to play an important role in the classification of both familiar, and similiar but unfamiliar, stimuli. Accordingly, they designed their study in a way intended to reveal the continuing effect of such individual episodes.

The material consisted of drawings of artificial 'animals' which differed along five 2-valued dimensions: body shape (rounded or angular), spots (present or absent), leg length (short or long), neck length (short or long), number of legs (2 or 6). The 'animals' were presented against one of four "ecological environments" (desert, forest, etc.), with each animal consistently associated with a particular environment. Animals were to be classified as either 'Builders' (i.e., living in shelters constructed from materials in their environment) or 'Diggers' (i.e., living in holes they dig). Half the subjects were told the classification rule at the start of the experiment ("rule subjects"), and half were not ("no-rule subjects").

Further details of the structure of the concepts and the design of the experiment are presented in the next subsection. However, there are a couple of points to note here about this original Allen & Brooks study:

- Complexities in the experimental design — of which there are several — are motivated (whether justifiably or unjustifiably) by the theoretical concerns of the study.
- It is clear from the authors' discussion (e.g. page 6) that they were not expecting the no-rule subjects to be able to learn the rule from the training undergone in the experiment. The authors' interest is in the subjects who were told the rule, as reflected in the title of the paper: "Specializing the operation of an explicit rule". The no-rule subjects are included as a control group to help answer specific questions about the data.

2.2 The Nottingham study

The study by McNaught & Gilmore (1996) appears to be an exact replication of the structure of the Allen & Brooks (1991) experiment, but with different experimental stimuli. Instead of artificial 'animals', McNaught & Gilmore use schematic drawings of 'airplanes' to be classified arbitrarily as either USA ('American') or AUS ('Australian'). The stimuli differ along five 2-valued dimensions: nose type (straight or dropped), shading (light or dark), wings (straight or swept), tail (high or low), and wheels (up or down). As in the Allen & Brooks experiment, four 'ecological backgrounds' were used: ground view at airport, looking down on airport, clouds, and bridge with cityscape.

Three of the five dimensions were relevant to the classification (nose, shading, and wings), while the other two were irrelevant. The classification was based on a 2-out-of-3 rule: if 2 or 3 of the relevant attributes had specified values then the plane was USA, otherwise it was AUS. (A little thought shows that the rule is symmetric: if less than 2 of the dimensions have their specified values, then at least 2 have the complementary values.) In the experiments, the actual stimulus values as to what counted as USA or AUS varied from subject to subject, e.g., whether it was shading light or shading dark, as part of the counterbalancing in the design. In this report (and in the modelling) we abstract from this variation by simply designating the dimension values as 0 or 1, giving us the classification rule:

Classification rule: an airplane is USA if at least 2 (i.e. either 2 or 3) of the dimensions nose, shading, and wheels have value 1; otherwise it is AUS.

Subjects in the experiment were given 40 training trials, followed by 40 test trials. In each training trial, subjects were presented with a stimulus, had to decide (or guess) its classification, and were then told the correct classification. The subject's latency and decision were recorded. The test trials were similar, but no feedback was given.

There are a number of complexities in the experimental design, the details of which are not all relevant to the present report but their flavour certainly is. For example, a subset of 8 of the 32 possible stimuli are chosen for use as the training set (see Section 4 below), so that over 40 training trials they are shown on average 5 times each. But the stimulus was chosen randomly and independently on each trial, so that by chance there might be 6 or 7 presentations of one stimulus and only 3 or 4 of another, and so on; and there was nothing to prevent the same stimulus being repeated on consecutive trials.

Another complexity concerns the composition of the stimuli used for the test phase (again, see Section 4). Stimuli in the test phase are divided into categories, referred to as *positive old*,

negative old, *positive match*, and *negative match.* It is not easy to understand these categories, though the description in the Allen & Brooks (1991) paper is marginally clearer than in McNaught & Gilmore (1996). The situation appears to be as follows. Because each stimulus may have either 2 or 3 relevant attributes that contribute to its classification, a second stimulus, which is similar to the first but differs from it on just one relevant dimension, may have either the same classification as the first (if they have between them 2 and 3 attributes with appropriate values) or the opposite classification to the first (if the first has 2 attributes with appropriate values and the second has just 1). Bearing that in mind, we have (Allen & Brooks, p.6):

- *Positive match:* an item seen for the first time in the test phase, which has the same classification as the corresponding similar old item.
- *Negative match:* an item seen for the first time in the test phase, which has the opposite classification to the corresponding similar old item.
- *Positive old:* an item seen in the training phase, whose corresponding similar new item is a positive match.
- *Negative old:* an item seen in the training phase, whose corresponding similar new item is a negative match.

Just to add to the general confusion, in the Nottingham experiment, because subjects had no basis for learning to classify the negative match stimuli correctly, their response was scored as correct if they gave the opposite answer to that based on the classification rule, i.e. if they gave the ‘wrong’ answer, and as an error if they gave the answer corresponding to the rule, i.e. a ‘right’ answer!

As in the Allen & Brooks experiment, half the subjects (N=20) were told the classification rule beforehand, and half (N=20) were not. In the present report we are concerned only with the no-rule subjects, who had to do their best to induce the classification from the training trials.

2.3 The basic concept-learning model

Young’s (1999) model of concept learning is simple and has a very open control structure. In other words, the model is not forced to “first do this, then do that”, but instead the actual trajectory of behaviour is determined by competition between production rules, and therefore depends upon learning.

The basic structure is shown in Figure 1, which shows the data flow (but *not* the control flow) of the model. The boxes hold production rules. There are 5 encoding productions, one for each dimension of the stimulus. Each production, if it fires, adds the value of its dimension to the description. There are 2 guessing productions, one to guess USA and one to guess AUS. Initially there are no classification productions, but they get learned from feedback during the training phase of the experiment. During training, each time a guess is made, i.e. a guessing production fires, a production is learned (or reinforced) associating the description with the correct classification.

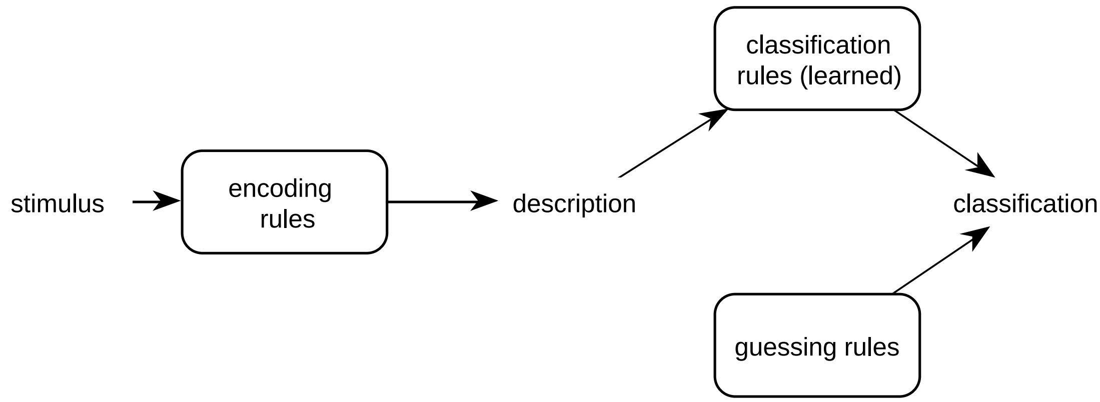


Figure 1. *Dataflow diagram of the Young (1999) model of concept learning.*

What is unusual about the model is that *all* the productions are in competition. The guessing productions compete with the classification productions, and the encoding productions compete with the guessing and classification productions. For example, it is possible for a guessing production to fire before any encoding production has had a chance to fire, thereby preventing any of the classification productions from firing on the basis of a description. Because in ACT-R productions compete on the basis of their past usage and history of success, and because the success of productions depends upon the information they need being present, the subtle interplay of competition and dependence between the productions can lead to complex long-term behaviour (Young, 1999). For present purposes however our focus is on the more direct consequences of the model's structure, such as that classification will often be made (i.e., a classification production will fire) on the basis of an incomplete description (i.e., before all the encoding productions have fired).

The model has been cast in this form specifically to allow for the possibility that it might learn to encode only the relevant dimensions. However, over the relatively short training runs encountered with the Nottingham dataset, i.e. just 40 training trials, this consideration does not come into play.

3. Preliminary analysis of model fit and variability

This section considers the basic fit between the model and the McNaught & Gilmore (1996) data, and examines the variability of the model in relation to that of the data.

3.1 Fit of model to Nottingham data

The primary measure from the Nottingham study is the percentage of stimuli correctly classified on the 40 test trials. Secondary measures break down this overall figure into figures for each of four subsets of stimuli.

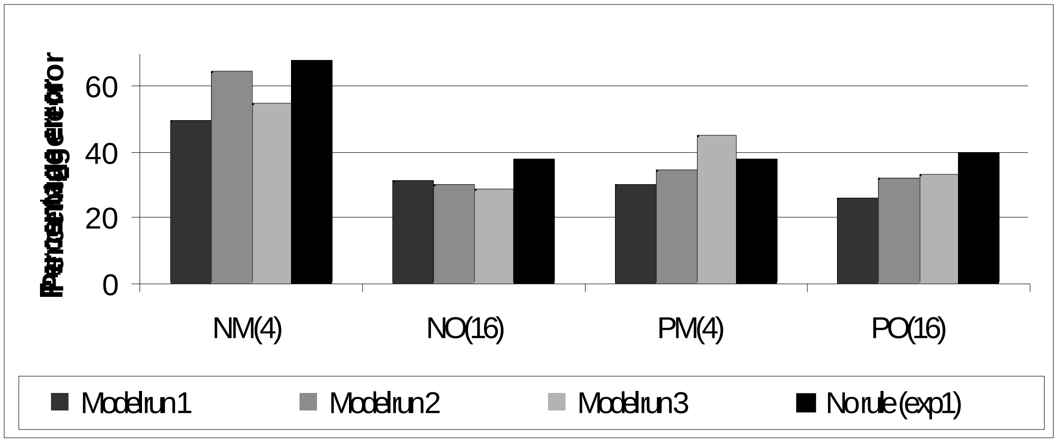


Figure 2. *Fit of 3 runs of the model against data from Experiment 1 of McNaught & Gilmore (1996).*

The overall success rate for the model is 65% and for the "no-rule" Ss (in Experiment 1 of the Nottingham data) is 50%. Figure 2 shows the results for the data and three runs of the model, for the four subsets of stimuli.

The model gives a fairly close fit to the data, without any adjustments to its parameters. We did not explore the possibility of optimising the fit by changing the parameters, as that seemed of little interest.

3.2 *Variability in the data*

One of the striking aspects of the Nottingham data is the high degree of variability among the Ss, much more than one would expect if the experiment were measuring an underlying "true" value (of percentage correct) with some random variation added. Figure 3 shows the overall performance of the 20 individual Ss in Experiment 2, computed from the Nottingham data. There is a very wide spread of results, with individual Ss making anywhere between 22% and 83% errors. It is notable that half the Ss are making more than 50% errors in a binary classification task! That hardly seems to indicate mastery of the task.

One route to understanding this variability is to model the task, and see if the model helps reveal the causes of the variability.

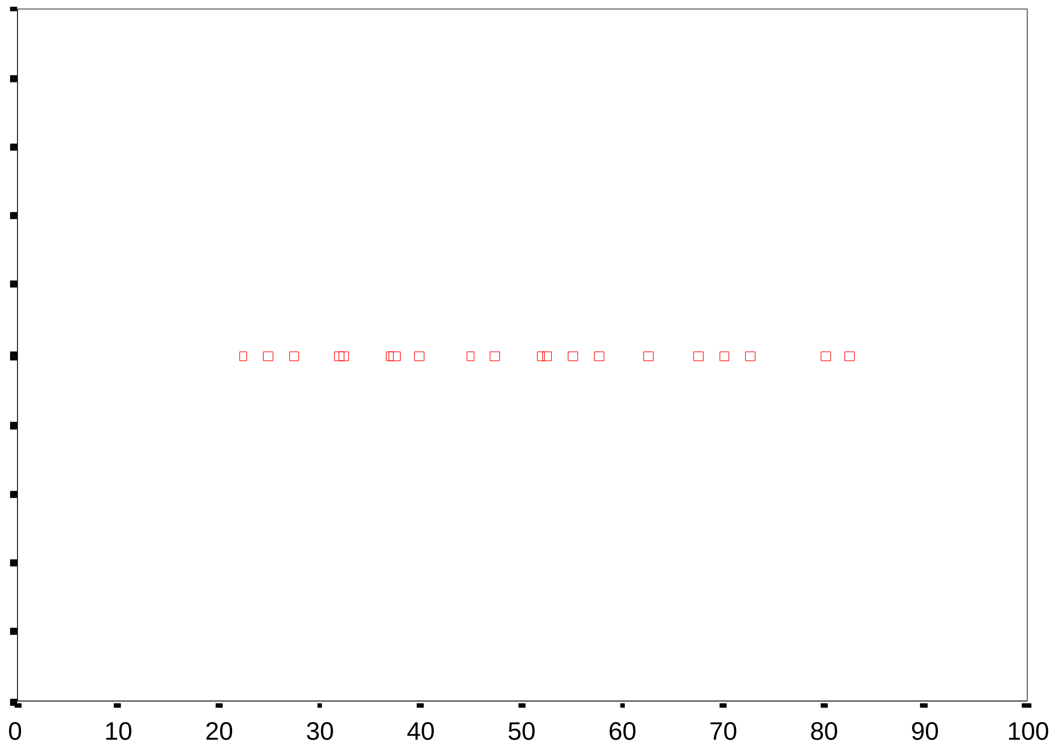


Figure 3. *Error rates of individual subjects in Experiment 2 of McNaught & Gilmore (1996).*

3.3 Variability in the model

For reasons that are a focus of the present study, the model too exhibits a high degree of variability from run to run. There are several random components to the model, so it is expected to show variability, however the degree of that variability still surprises us, and is clearly indicative of something inherent in the structure of the model.

Figure 4 shows the distribution of error rates for 9 separate passes of the model. For each pass, the model was run with the exact sequence of training and test stimuli as were experienced by each of the 20 Ss. In other words, each row corresponds to the model "simulating" each of the 20 Ss in turn, in the sense of receiving and responding to the identical sequence of inputs, as did the S.

The horizontal scale in Figure 4 is not directly comparable to that of Figure 3, because Figure 3 refers to data from Experiment 2, whereas in Figure 4 the model is simulating Experiment 1. (Experiment 2 included a sixth attribute with 4 possible values, redundant with the other attributes.) However, it can be seen that the individual points, each corresponding to the

simulation of an individual S, yield a wide spread of performance similar to that in Figure 3. Here, the error rates range from about 10% to about 60%.

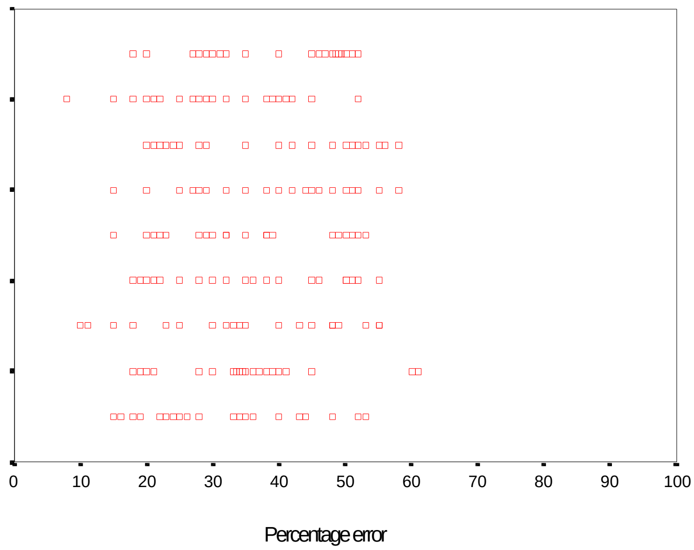


Figure 4. *Nine passes of the model, showing individual results for each of the 20 training and test stimulus sets.*

3.4 Variability due to stimulus sequence

Understanding the causes of this variability is a major focus for the project. One possible factor in the variation between human Ss lies in the design of the experiment, where the randomisation of stimuli means that the order in which stimuli are encountered, and even the exact numbers of stimuli of different kinds, vary from one S to another. It is impossible to assess the effect of this factor in the experiment, where of course each S is tested only once, and therefore encounters once a unique sequence of stimuli. But we can gain some leverage on the question by testing whether the stimulus sequence has a systematic effect on the performance of the *model*, which of course we can run repeatedly in order to generate the data required for statistical analysis.

With the help of a colleague, Dr Diana Kornbrot who is a mathematical psychologist, we applied a logistic analysis to the data shown in Figure 3, which is roughly analogous to an Analysis of Variance but appropriate where the data represent proportions — here, the percentage of errors. We regarded the 20 different stimulus sequences (each being the sequence received by one of the human Ss) as an independent variable with 20 levels. To understand the question being asked here, it is helpful to focus momentarily on say the leftmost point in the bottom row in Figure 4. For that run of the model, the stimulus sequence is identical to that received by one of the Ss, and it led to a comparatively low error rate. The question being asked in the statistical analysis is whether there is a systematic tendency for that particular input sequence to lead the model to a

low error rate in other passes too, for example in the next pass (second row in the figure), the following pass (third row), and so on.

The analysis shows that there is indeed a systematic contribution from the input sequence, and that it accounts for around 5%-9% of the variability, depending on just how it is measured. So we get an interestingly mixed answer to our question. Part of the variability between Ss can be attributed to the individual, idiosyncratic stimulus sequence that each encountered. However, most of the variability must be due to other causes.

3.5 Extension of model to Experiment 2

The ACT-R model of concept classification (Young, 1999) that we were working with corresponds to Experiment 1 of the Nottingham study (McNaught & Gilmore, 1996), whereas the data from the Nottingham study that we have comes from their Experiment 2. Experiment 2 (based on the original Allen & Brooks [1991] paper) adds a sixth, but redundant, attribute to the stimuli, consisting of one of four different backgrounds. Of the 8 stimuli used during training, two are associated with each of the backgrounds, one being a USA plane and the other an AUS plane, so that the background is an 'irrelevant' attribute.

The model was extended, straightforwardly enough, by adding a sixth, 4-valued attribute to the five 2-valued attributes already present. Making this change did not improve the fit of the model to the Experiment 2 data (compared with the fit of the model to Experiment 1 data). Although one can fiddle with details of the model in an attempt to improve the fit, the main problems appear to lie with anomalies in the data, for example where two groups of stimuli which ought to have been equivalent from the Ss' point of view yielded markedly different error rates (50% versus 20% for the positive olds and negative olds respectively).

4. Flaws in the experimental studies

Further experimentation with the salience of different encoding rules, e.g. having a different salience for relevant as against irrelevant attributes, led us to examine carefully the pattern of attributes in the stimuli being shown to the subjects and the model. We had been aware from the beginning that the design of the McNaught & Gilmore (1996) study, identical to that in the Allen & Brooks (1991) paper, was a little eccentric, but just how eccentric we had not yet appreciated.

If we look first at the range of stimuli used, we find that of the 32 possible stimuli, just 8 are presented during the 40 trials of the training phase, being shown on average 5 times each. During the 40 trials of the test phase, those same 8 stimuli are used for 32 of the trials (being shown on average 4 times each), mixed in with 5 other items, 3 of which are shown twice and 2 once. In other words, all the training and the majority of the testing is done with just 8 of the possible stimuli. Given that those stimuli are shown several times each, the study would appear to be as much about learning the classification of those particular stimuli as about the acquisition of the general concept.

There is worse news if one examines the pattern of attributes occurring in those 8 stimuli. Table 1 shows the stimuli used for the training (and most of the testing) of a particular subject, A421.

	'Relevant'			'Irrelevant'			
Plane	Nose	Shading	Wings	Tail	Wheels	B/gnd	Class
2	0	1	1	0	1	1	USA
6	0	1	0	1	1	1	AUS
3	1	0	1	1	1	2	USA
5	1	0	0	0	1	2	AUS
4	1	1	0	1	0	3	USA
8	0	0	0	0	0	3	AUS
1	1	1	1	0	0	4	USA
7	0	0	1	1	0	4	AUS

Table 1. *The stimuli used for training subject A421 in Experiment 2 of McNaught & Gilmore (1996).*

Supposedly, as shown in the table, the Nose, Shading, and Wings are relevant attributes, in the sense that two or more of them having value '1' means that the plane is USA, two or more with value '0' means AUS; while Tail, Wheels, and Background are irrelevant in the sense that their value does not affect the classification of the stimulus. However, as can be seen from the columns for Tail and Background (shaded in the table), it happens that *for these particular stimuli,* each of the four backgrounds occurs with a '1' for the Tail and with a '0' for the Tail. Consequently, although neither the value of the Background nor the value of the Tail attribute by themselves provide information about the classification of the stimulus, their combination certainly does. In fact, together they serve to identify a unique plane. For example, plane #2 (on the first line of the table) is identified by Background=1 and Tail=0, while plane #7 (on the last line of the table) is identified by Background=4 and Tail=1. This means that the classification of the 8 stimulus planes can be based as easily on two of the supposedly *irrelevant* attributes as it can on the three *relevant* attributes — or perhaps even more easily, given that only 2 attributes need be considered instead of 3, one of them being the most salient!

Because of this unclarity about what counts as a relevant attribute, as well as other features of the experimental design that border on the bizarre, we regard it as pointless to try to interpret findings from the experiment at face value, or to attempt to tune the model to be a close match to the data. (For example, in addition to the peculiarities of the experimental design already

mentioned, the McNaught & Gilmore report talks of subjects being dropped from the analysis for obscure reasons, in one place casually mentioning that 26 of the 40 subjects were thus dropped.) For future work, instead of treating the model as an 'artifical subject' in the experiment and exposing it to the same sequence of stimuli as real subjects, we will simply train and test the model on blocks of 32 trials, each of which is a random permutation of the 32 possible planes.

5. Random walk analysis of variability

Early in the project, we came to realise that the Young (1999) model is very similar to an existing concept-learning model in ACT-R, Niels Taatgen's model of Kendler & Kendler's (1959) experiment in simple concept learning in children, described in Anderson & Lebière (1998). The main difference is that in our model, the encoding productions compete with the classification and guessing productions, whereas in Taatgen's Kendler & Kendler (K & K) model the classification stage is delayed until exactly one of the encoding productions has fired. Furthermore, Taatgen's productions for guessing are arranged to fire only if there are no applicable classification productions. Thus, the anarchic competition between productions which makes the behaviour of our model so complex and difficult to analyse is absent from the K & K model. In the K & K model, the principal competition is between the two encoding productions as to which dimension to encode. We already know that the K & K model exhibits a wide range of variability, so the possibility of analysing the sources of variability in the K & K model provides potential leverage in understanding our own model. The variability shows up in Table 2, which displays the standard deviation of the mean difference between the model and the data averaged over *N* runs, for various values of *N*.

N	Mean no. of trials	SD of mean
5	18.19	4.01
10	17.06	2.95
20	16.82	1.98
50	17.18	1.40
100	16.95	0.88
200	17.01	0.62
500	17.16	0.41

Table 2. *Standard deviation of the root-mean-square difference between model and data for the non-reversal ('extra-dimensional') shift in the Kendler & Kendler experiment averaged over N runs. (SD based on a sample of 50 in each case.)*

To check on the underlying cause of this variability, we ran the K & K model a large number of times to observe the distribution of non-reversal learning times. The results are shown in Figure 5 as a histogram over 100,000 runs. It can be seen that although the bulk of the learning times form a reasonably compact distribution, there is a long tail which stretches on and on. Although the mean value is around 17 trials to learn, there are occasional runs in the 80s and 90s.

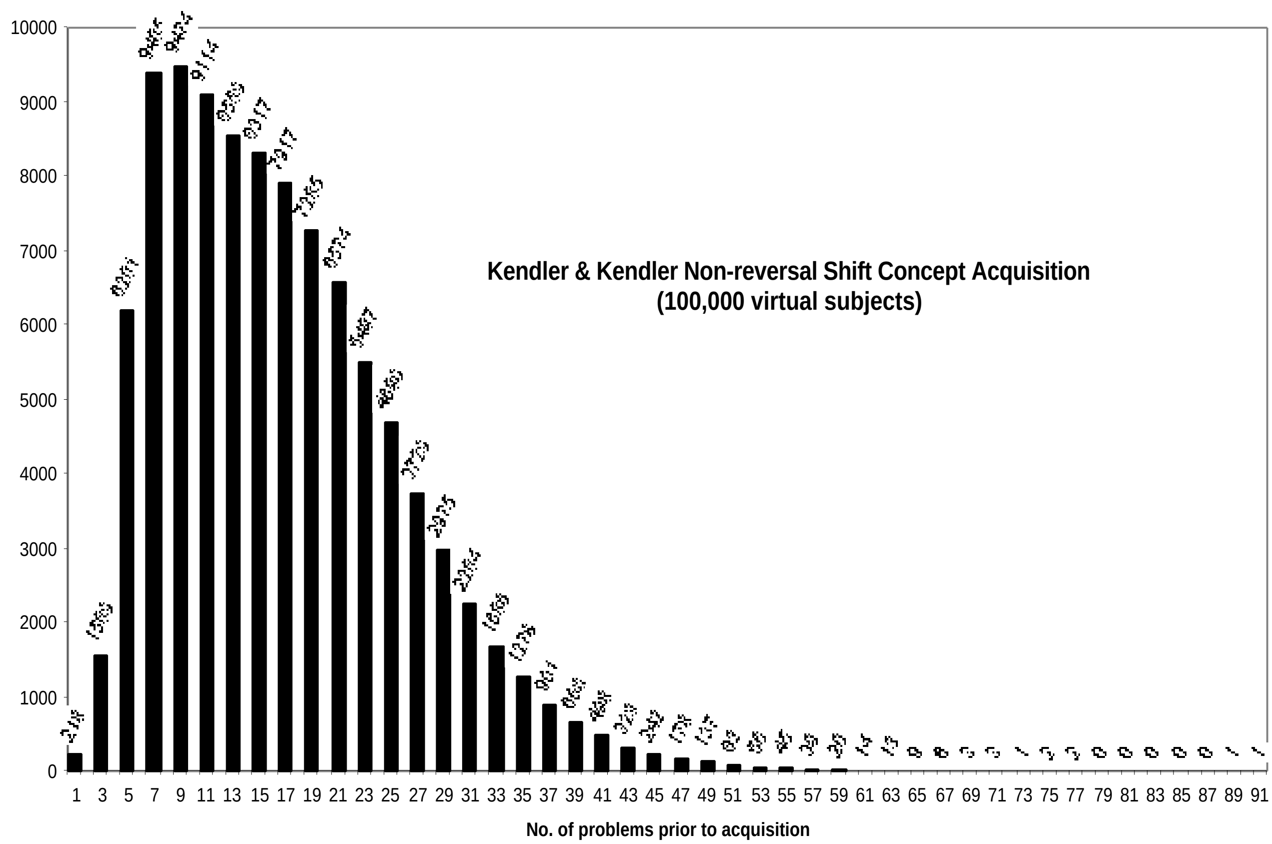


Figure 5. *Distribution of 100,000 non-reversal learning times in the Kendler & Kendler model.*

In trying to work through why the K& K model sometimes exhibits much longer learning times than average, we realised that part of the ACT-R mechanism involves a process recognisable as a *random walk.* In probability theory, a random walk is a multi-step stochastic process in which at each step, a variable either increases or decreases in value. For example, a basic *symmetric random walk* (Feller, 1957) has a variable which at each step either increases or decreases by 1 with equal probability. A representative random walk is shown in Figure 6, where a variable begins with value 0 which randomly increases or decreases by one at each step. With time, i.e.

with increasing number of steps, the variable tends to drift further and further away from its initial value. Some of the interesting questions about random walks concern their properties of *first passage.* Figure 6, for example, shows barriers at ±5. One can ask about the probability that the random walk will reach either +5 or –5 within N steps, or about the distribution of the number of steps it takes to reach one of those barriers.

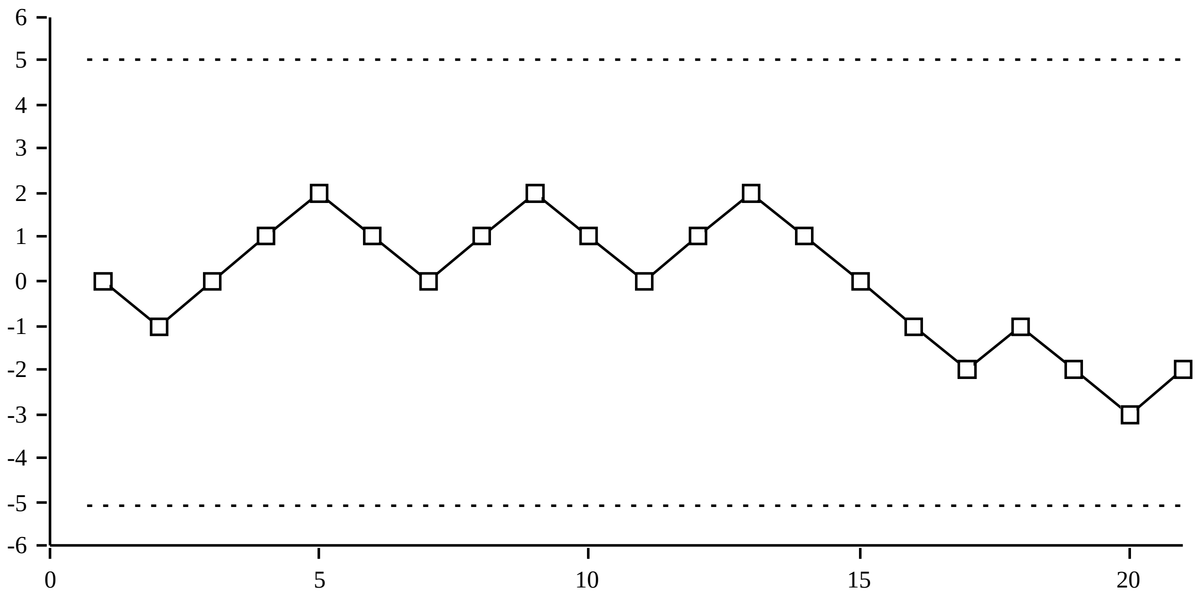


Figure 6. *The first 20 steps of a random walk, where the value starts at zero and increases or decreases by one at each step.*

Without going into technical details (see, e.g. Feller, 1957), the main relevance for our present concerns is that the distribution of times to first passage has a very long tail. In Figure 5, suppose we are interested in how long the walk will take to reach the +5 barrier. In the sequence shown, the walk begins by moving towards the upper barrier, then drops and approaches the lower barrier. If the walk were to continue, it might continue downwards for a few more steps, and then turn upwards again and take another 20-30 steps to reach the upper barrier. The times can be counter-intuitively long. The modal (i.e,, most common) time to reach the +5 barrier is 7-9 steps, but the median time is 53 steps, the third quartile (i.e. covering 75% of the cases) is not reached until 245 steps, while the 90-percentile is greater than 1200. In other words, 10% of the walks take more than 1200 steps to reach the +5 barrier. In fact, the distribution is so extreme that the average number of steps to reach the barrier is technically infinite.

The connection with our concept learning models hinges on the observation that a critical part of the learning depends upon waiting for a random walk to cross a barrier. This is easiest to explain for Taatgen's K & K model. The Kendler & Kendler (1959) task uses concepts with two 2-valued attributes: *size* (large or small) and *colour* (red or green), only one of which is relevant.

As in our own model, there is a production to encode each attribute, but unlike our own model, the model is constrained to use one or other of those two encoding productions. ACT-R's learning mechanism is relied upon to discover which of those productions is correct, i.e., which one leads to success. The choice of which production to apply depends upon their relative histories of success in the past, specifically upon the ratio of the number of occasions on which each production has been successful to the number of times it has been used. Initially in the experiment, colour is the relevant attribute, and at the point where the concept is declared to be "learned" the encoding production for colour has a success ratio of around 0.95 (i.e. it has led to success on around 95% of the occasions on which it has been applied) while the encoding production for size might have a success ratio around 0.85. That difference between the ratios is sufficient to ensure that colour is encoded in preference to size virtually all the time.

At this point in the experiment, without warning, the concept is changed such that the colour attribute becomes irrelevant and the size attribute becomes relevant. Now, if the model were able to fire the size-encoding production, it would rapidly learn that encoding size leads to consistent success. But as things stand it cannot fire that production, because the success ratios mean that the colour-encoding production is (almost) always chosen. We have to wait until the ratio for the colour production has fallen to near the ratio for the size production before size gets a chance to be encoded. This will certainly happen sooner or later, because the colour-encoding production, now being irrelevant, has only a 50% success rate, so in the long run its ratio will drift down towards 0.50. However, because it is still successful on half the occasions it is used, the process of changing its success ratio from 0.95 to somewhere near 0.85 takes the form of a random walk, and because of the properties of random walks that change can sometimes take a long time. Although the typical time to learn the new concept might be around 17 trials, occasional runs may take 50 trials, or 100 trials, or even longer.

We can understand more about these very long learning times by examining an individual run in detail. Figure 7 depicts a run where the model took 94 trials to learn after the non-reversal shift. The figure plots the expected gain, E, for the two encoding productions, *attend-color* and *attend-size,* where E is calculated by the formula PG – C, with P being the probability that the production will lead to a correct classification, G being the value of the goal (conventionally set at 20 units), and C the cost of reaching the goal via this production firing. In these cases, the variable P simply corresponds to what we above called the 'success ratio', the proportion of the occasions it has been fired that have led to success.

Initially, up till around cycle 25, only the *attend-color* production fires. Because it is successful only about half the time, its P value — and hence its E value — drifts downwards. Once its E is reasonably close to that for the *attend-size* production, *attend-size* gets a chance to fire (on cycles 26, 31, 42, 57, and 86). Finally, from cycle 94, *attend-size* begins to fire every time and the new concept is learned. From the plot, we can identify two factors that contribute to the length of the run. One is that *attend-color* has two spurts of "good luck" where it happens to lead to several correct answers in a row (around cycles 4-12 and again in the 30s). The other is that the second production, *attend-size,* experiences some mild "bad luck", in that on each of the first four times it fires, it leads to a wrong answer, which further lowers its E value and means that *attend-color* has to fall still further before *attend-size* gets another chance to fire. So in this case, we can see

that a particularly long learning run is due to a combination of random walk characteristics with some individual instances of good and back luck.

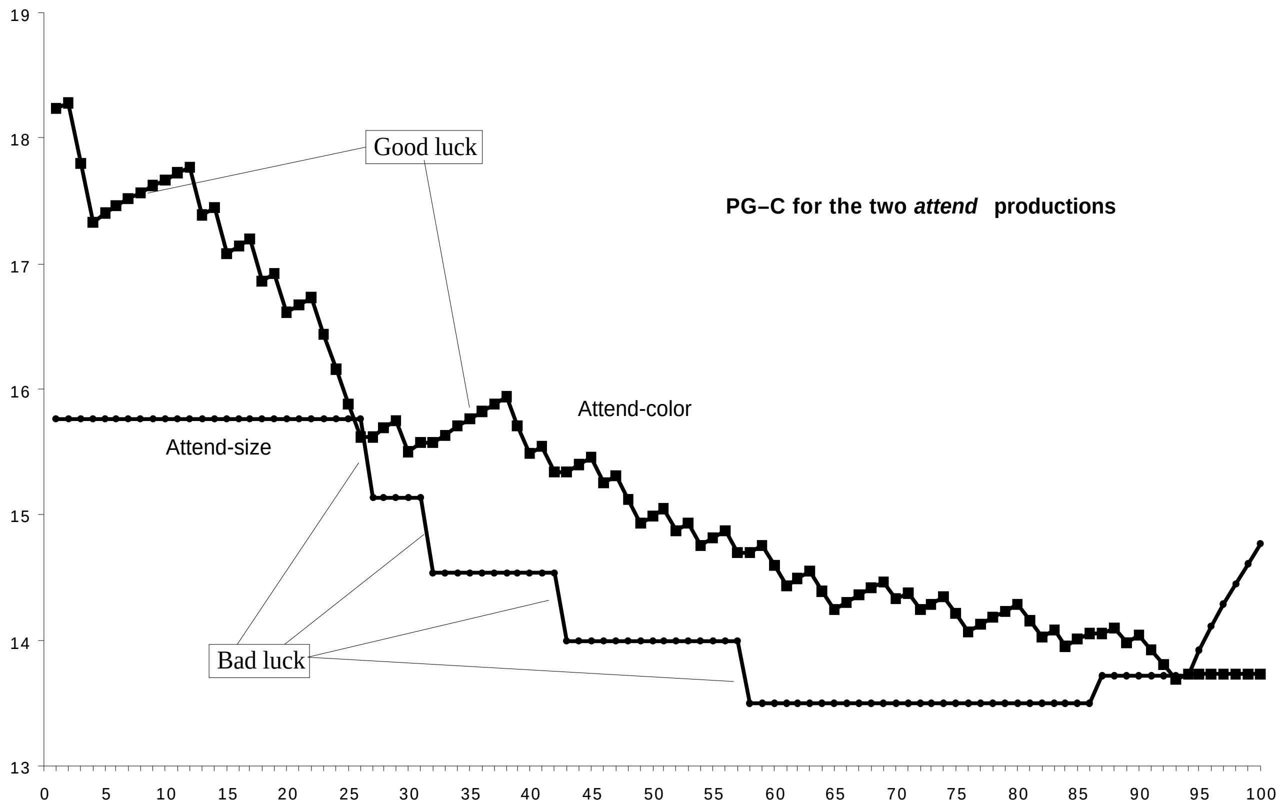


Figure 7. *The expected gain (E = PG – C) for the two encoding productions in the Kendler & Kendler model, during an unusually long run for non-reversal learning.*

In our own model for plane recognition, there is nothing corresponding to the K & K non-reversal shift to set up an over-dominant production which has to be weakened by a random walk. Nonetheless, the K & K model is relevant because it illustrates also how a production can become temporarily dominant by a series of chance events. Figure 8 shows a plot analogous to that of Figure 7 except that it occurs during the initial learning on the K & K task. Initial learning typically takes 6-7 trials, but on this run requires 26 trials. Figure 8 shows why. Although *attend-size* is the irrelevant encoding production, it happens that on the first eight occasions when it fires, it sees a green stimulus. It therefore learns to classify small green stimuli correctly as “no” on trial 2 and uses that learning to classify them correctly on subsequent trials 5 and 8, and similarly learns to classify large green stimuli as “no” at trial 6 and applies that learning on trials 7, 11, 12, and 14. It is only when *attend-size* fires for a red stimulus on trial 15 that it loses out finally to *attend-color*. Even then, it fires once more by chance on trial 25,

thereby delaying the final learning to criterion by a further 9 cycles. So, for trials 5-14, *attend-size* is on a random walk that keeps it above the other production.

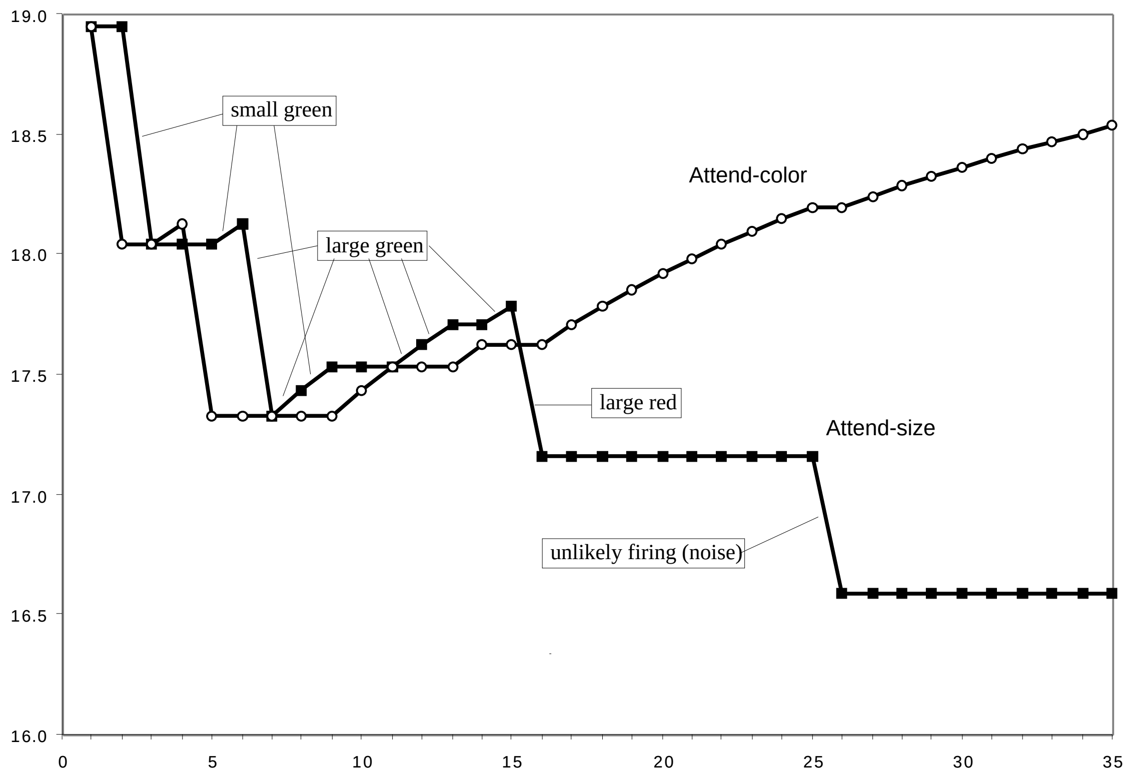


Figure 8. *The expected gain (E = PG – C) for the two encoding productions in the Kendler & Kendler model, during an unusually long run of initial learning.*

There is a bit more to the story, some of it still to be worked out. However, we are offering this explanation as an answer to the primary question which drove this research project: why there is so much run-to-run variability in the behaviour of models of the class we are considering. Clearly, more investigation is needed. There are also interesting empirical questions to ask about whether the distribution of learning times in human subjects fits the distribution predicted by our analysis. If that were to be the case, it would constitute a very stringent test of the ACT-R learning mechanism.

6. Representation and coverage of concepts

The supposed scientific motivation for the concept learning experiments behind this study (Allen & Brooks, 1991; McNaught & Gilmore, 1996) is to address the question of how people represent

the kind of concepts we are dealing with here. The studies are designed to address the theoretical question of whether people represent such concepts in the form of "rules or instances". Our cognitive modelling stance provides a helpful perspective on the question because it both concretises and cuts across the dichotomy of 'rules' and 'instances'.

The model provides an illuminating perspective on the theoretical question, because it contains aspects of both possibilities. The model as it stands certainly never learns a "rule", in the sense of learning a formula such as "If at least two of the relevant attributes have such-and-such values, then the plane is USA, else AUS". On the other hand, neither does it store individual, complete instances. What it does instead is to learn and store a considerable number of partial patterns (or specifications) of the USA or AUS concepts, of varying degrees of completeness, each derived from the encounter with a particular instance.

As our model runs, it gradually builds up a representation of the concept in the form of rules, each of which associates a description of a subset of the possible stimuli with its classification as USA or AUS. Those descriptions vary from fully specific, i.e. applicable to just one single stimulus, to extremely general, such as "any plane with Shading=dark", and therefore applicable to half of all the stimuli. As well as varying in specificity, which affects the degree of coverage of the stimuli, the rule descriptions also vary in their *relevance*, the extent to which they encode relevant (as against irrelevant) attributes of the stimulus. For example, a rule which describes two attributes (and therefore covers a quarter of the stimuli) might specify the values of two relevant attributes, or two irrelevant attributes, or one of each.

We found this all a bit confusing to think about. Although we had a vague mental picture of the concept gradually building up in the model as a collection of production rules which together 'cover' more and more of the concept, the variations among rules in specificity and relevance, and the fact that rule descriptions can overlap in different ways and to different degrees, made it difficult for us to achieve a clear understanding. Accordingly, we devoted some effort to clarifying this issue, and we also began to develop a graphical notation to help us visualise and track the gradual 'growth' of a concept within the model.

To describe the notation, we first summarise the task, introduce some distinctions, and do some counting:

- We will talk of 'planes' rather than 'stimuli'.
- Planes are represented as five binary attributes. Three of those attributes (nose, shading, wings) are relevant to the plane's classification as USA or AUS. The other two attributes (tail, wheels) are irrelevant.
- Rather than deal with concrete attributes and values, such as "shading = dark", we will abstract to R1, R2, and R3 for the relevant attributes and I1 and I2 for the irrelevant. The values will be taken as 1 for USA and 0 for AUS.
- The classification follows a "2 out of 3" principle: if 2 or more of R1, R2, R3 have value 1, then the plane is USA; otherwise 2 or more of them have value 0 and the plane is AUS.
- Since there are five binary attributes, the number of different planes is $2^5 = 32$.

- The number of possible descriptions is quite different. In a description, each attribute may have value 0, or value 1, or not be included. Hence, the number of possible descriptions is $3^5 = 243$.
- Each rule associates a description with a classification. Since there are 243 possible descriptions and 2 possible classifications, the number of possible rules is 243 × 2 = 486. (In fact, not all of those rules can ever be learned. The rules which are always wrong can never be acquired. There are 72 such rules, so the greatest number of rules that can exist in the model is 486 – 72 = 414.)

Next, descriptions, and their use in classification rules, fall into classes depending upon the number of relevant attributes included in their description:

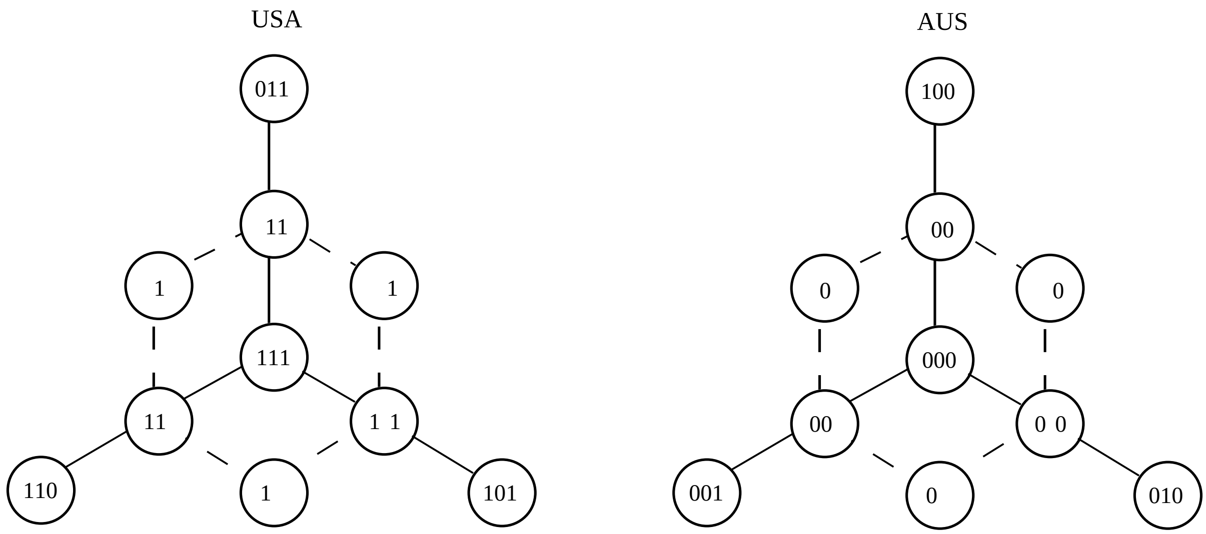
- A 3R rule, i.e. encoding all three relevant attributes (and any number of irrelevant ones) always gives a correct classification, so we also think of it as of type R-100, because it is correct 100% of the time.
- There are two varieties of 2R rules. A description which encodes two relevant attributes with the same value (i.e. both 0 or both 1) will be correct 100% of the time, and so is also a R-100. A 2R rule which encodes different values (e.g. R1=1 and R2=0) can be correct only 50% of the time and therefore does not contribute to the representation of the concept, since it yields no improvement over guessing.
- A 1R rule, i.e. with just one relevant attribute, will be correct 75% (or 25%) of the time. For example, a rule which classifies R1=1 as USA will be correct for 3 of the 4 possible values of R2 and R3, and will be incorrect only for the case R2 = R3 = 0. This R-75 also provides useful coverage of the concepts.
- All other rules are correct 50% of the time, and therefore do not contribute to the representation of the concept.

We took a big step towards a coherent view of the model when we realised that the coverage of its representation must be thought of against the 243 possible descriptions rather than against the 32 planes. After all, the classification rules never ‘see’ the actual plane. Instead, the plane is presented to the model; then zero, one, or more encoding rules fire; and then a classification rule. So the model really has to learn to classify *descriptions*, not *planes*.

6.1 Depicting the relevant attributes

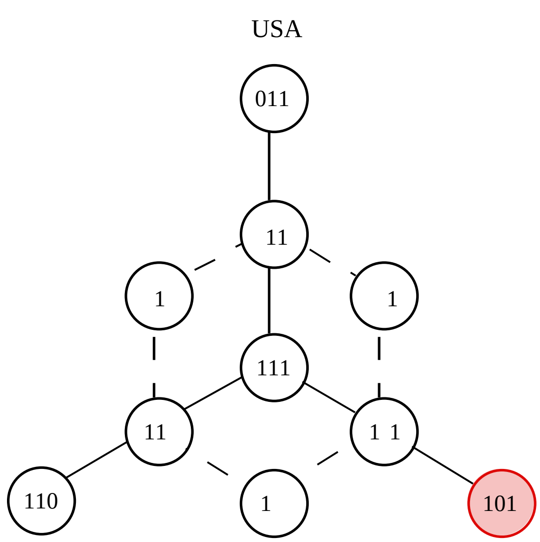
The next move is to form a graphical notation for the descriptions — or at least, for those descriptions which form part of the concept representation — on which we can visually depict the growing coverage of the concept by the model. We begin by focussing on the Rs (relevant attributes): for the moment, we simply ignore the existence of the Is (irrelevant attributes).

In the descriptions, each of the Rs has three possible values: 0,1, and ‘unencoded’ which we will denote by ‘ ’. The possible descriptions can be thought of as lying on a 3 × 3 × 3 cube, with one vertex being USA (R1=R2=R3=1), and the opposite vertex being AUS (R1=R2=R3=0). Rather than working with the cube itself, we draw two projected views of it, one centred on the USA vertex and one on the AUS vertex. (Fortunately the only interior point, R1=R2=R3= , i.e. the null description, is not part of the concept representation.)

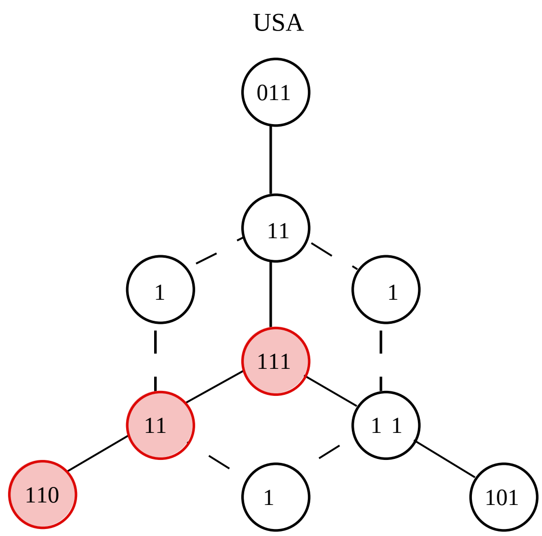


The left hand part of the diagram shows the space within which the concept of “USA plane” exists. What we now do is to begin “colouring in” the diagram as classification rules are learned by the model.

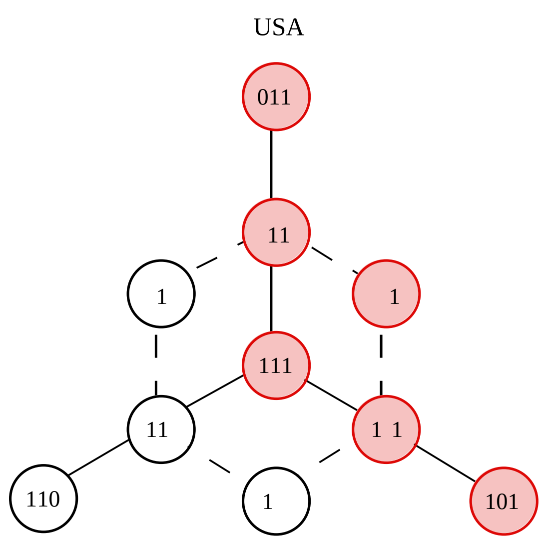
- A 3R rule, i.e. one which describes values for all three of the Rs, covers just a single node in the diagram. For example, if the model learns the rule “101 USA” (in other words, the classification rule “If R1=1 and R2=0 and R3=1 then classify as USA”), then we colour in the 101 node:



- A 2R-100 rule, i.e. one which describes two of the Rs with the same value, covers a whole edge of the cube. So the rule “11 USA” would colour in all three nodes on an arm of the diagram:



- A 1R-75 rule, i.e. one which describes a value for just one of the Rs and gives the corresponding classification, covers a whole face of the cube and therefore colours in the nodes in one of the three ‘sectors’ of the diagram. For example, the rule “ 1 USA” would colour in these nodes:



As rules are aquired one by one, we continue to colour in the diagram. With all three rules shown in the examples above, all the nodes in the USA space would be coloured apart from the 1 and 1 nodes. Some of the nodes (e.g. 111 and 101) are covered by more than one rule. If two or more further rules were acquired that covered the 1 and 1 cases, then all the USA nodes would be covered, and the model would essentially ‘have’ the concept of a USA plane.

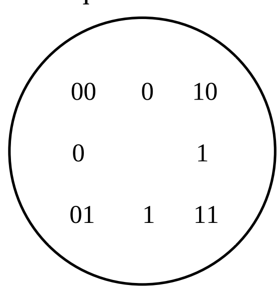
A similar story holds of course for the AUS side of the diagram.

It should be noted that there are many different ways in which the nodes can be covered, and the model thereby ‘have’ the concept. At one extreme, there could be a lot of 3R rules, each covering just a single node; or by fewer 2Rs, each covering an edge of the cube; or by some 1Rs, each covering a sector. Typically, however, there would be a combination of the different types.

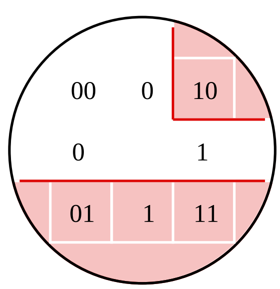
The 1R (R-75) rules introduce the possibility of error, since 25% of the times such a rule can fire it would wrongly classify an AUS plane as USA. For example, the “ 1 USA” rule shown above could fire for a plane described as 001 which is in fact AUS. However, such errors will be relatively rare in a fully trained model, since there is a good chance that the 001 description would trigger an R-100 rule to classify the plane (correctly) as AUS, and after training an R-100 rule will usually beat an R-75 rule.

6.2 Depicting the irrelevant attributes

We now complete the story by explaining how the irrelevant attributes are taken into account and handled in the graphical notation. To include the irrelevant attributes (abstractly labelled as I4 and I5), each of the nodes shown as a circle in the diagrams seen so far — representing a particular point in the space (cube) of possible descriptions involving the relevant attributes R1, R2, R3 — is itself divided into a 3 × 3 square of 9 cells representing the values of I4 and I5:



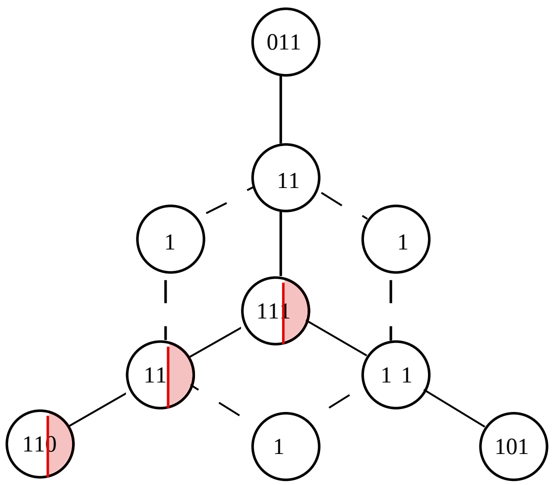
For each node covered by a rule, instead of the whole circle being coloured in as we have done so far, just the parts corresponding to the values of I4 and I5 get coloured in. For example, if we have I4=1 and I5=0, then we colour just the upper right quadrant. If the rule specifies just I5=1 (and hence I4=), then we colour the lower segment (slightly less than a semicircle). After making both colourings, we have:



Only if there are no irrelevant attributes specified (i.e., I4 = I5 =) do we colour in the whole circle. Interestingly, this is the only way to get the whole node covered, since otherwise any combination of values for either or both of I4 and I5 will leave the central, ‘ ’, cell uncoloured.

The actual colouring-in done to represent a rule has to reflect both the relevant and irrelevant attributes included in the description. For example, the if the description has R1=1, R2=1, and I4=1 — what we might write as ‘11 |1 ’ — then for each of the three nodes on the edge corresponding to R1=1, R2=1, we colour in the right-hand segment, corresponding to I4=1:

USA



6.3 Using the notation

So the graphical notation is used as follows. Initially, the diagrams for USA and AUS are uncoloured. As classification rules are learned, we colour in (parts of) the corresponding nodes. Gradually, the colouring-in gets more and more complete, i.e., more and more of the USA and AUS nodes are more completely coloured in. At any stage, the extent to which the nodes for USA or AUS are more or less completely coloured represents the extent to which the model ‘has’ the corresponding concept.

It should be noted that we are not claiming any substantive theoretical contribution from the work reported in this section. Its value is primarily heuristic, in helping us to clarify our thinking and understand what the model is doing. It is also highly specific to the particular materials we are dealing with. However, the ideas may have some practical value, perhaps in a training context, in tracking a trainee’s progress with acquiring a skill.

References

Allen, S. W. & Brooks, L. R. (1991). Specializing the operation of an explicit rule. *Journal of Experimental Psychology: General, 120* (1), 3-19.

Anderson, J. R. & Lebière, C. (1998) *The Atomic Components of Thought.* Erlbaum.

Baxter, G. D. (1997) SCA-PR. Software (a plane recognition model written in Soar). Psychology Department, University of Nottingham, UK.

Feller, W. (1957) *An Introduction to Probability Theory and its Applications.* Volume 1. Wiley.

Kendler, T. S. & Kendler, H. H. (1959). Reversal and nonreversal shifts in kindergarten children. *Journal of Experimental Psychology, 58,* 56-60.

McNaught, D. E. & Gilmore, D. J. (1996) Rules or episodes in rapid decision making? Unpublished draft paper. Psychology Department, University of Nottingham, UK.

Miller, C. S. & Laird, J. E. (1996) Accounting for graded performance within a discrete search framework. *Cognitive Science, 20,* 499-537.

Newell, A. (1990) *Unified Theories of Cognition.* Harvard University Press.

Oni, V. I. (1998) An experimental comparison of concept acquisition between a symbolic-based model and the corresponding human data. Unpublished Honours project in Cognitive Science. Psychology Department, University of Hertfordshire, UK.

Young, R. M. (1999) Long-term learning in a simple, dynamical, self-organising ACT-R model of concept acquisition ... or, Why individual Ss vary so much. Unpublished talk presented at ACT-R Summer School, Carnegie Mellon University, Pittsburgh.

Appendix: ACT-R Code for the Concept Learning Model

```
;;; Concept acquisition model

;;; Started 29.7.99

;;; Version 3

;;; The big change now is that rule learning takes place ONLY after guessing.
;;; The rationale is that if a learned classification rule gives the wrong
;;; answer, then it is an unreliable rule, and one doesn't want to acquire its
;;; converse, which will be equally unreliable.

;;; As a guide to setting the parameters, we need to know the "true" Ps of
;;; different rules:
;;;   Guessing rules should stay close to 50%
;;;   Encoding rules will basically match the overall success rate, which
;;;      begins at a little over 50% and hopefully drifts upwards.
;;;   Classification rules are mostly 50% (if irrelevant) or 100% (if correct),
;;;      with occasional 75% (if one attribute correct).  Note though that all
;;;      new rules start at 100%, and become either 50% or 100% after first
;;;      firing.

;;; The policy at the moment is to set the Encodings initially a bit above 50%,
;;; say around 60%, with enough noise that do get some early guessing but only
;;; occasionally.

;;; There is a slight complication with learned "premature guessing" rules,
;;; i.e. rules which ignore all attributes and just give a response.  They can
;;; be a real nuisance, because they can accidentally get high priority and
;;; then tend to block all further learning.  (They could be prevented, but
;;; not really worth the trouble since get essentially the same problem with
;;; (most) 1-attribute rules.


;;; ================================= LISP STUFF ===========================

;;; MUCH OF THE LISP CODE HAS BEEN SUPPRESSED
;;; Full code is available on request from the author:
;;;     Richard M Young <r.m.young@herts.ac.uk>
;;; ......

;;; Call e.g. (setup-classify-one '(nose square shading light ...) 'AUS)

;;; (setup-classify-one '(nose square shading light wings swept tail large wheels up)
'USA)

(defun setup-classify-one (properties response)
   (setq true-class response)
   (delete-chunk the-stimulus)
   (eval (list 'add-dm (append '(the-stimulus isa the-stimulus) properties)))
   (if (no-output (dm maingoal))
       (mod-chunk maingoal nose unencoded shading unencoded wings unencoded
              tail unencoded wheels unencoded class nil)
       (add-dm (maingoal isa classify nose unencoded shading unencoded wings unencoded
              tail unencoded wheels unencoded class nil)))
   (goal-focus maingoal))

(defun classify-one (properties response)
   (setup-classify-one properties response)
   (run))
```

```
;;; ......

;;; Instead of running repeatedly through the A421 stimulus set, make multiple
;;; passes through a randomized presentation of all 32 possible stimuli.
;;;
;;; For the record, the stimuli and values are:
;;;    nose = down | square
;;;    shad = dark | light
;;;    wing = swept | square
;;;    tail = small | large
;;;    whee = up | down
;;;
;;; The rule is R1: nose down, shad dark, wing swept

(defun one-pass ()
   (let ((stimlist
           (randomize
             '((down dark swept small up) (down dark swept small down)
               (down dark swept large up) (down dark swept large down)
               (down dark square small up) (down dark square small down)
               (down dark square large up) (down dark square large down)
               (down light swept small up) (down light swept small down)
               (down light swept large up) (down light swept large down)
               (down light square small up) (down light square small down)
               (down light square large up) (down light square large down)
               (square dark swept small up) (square dark swept small down)
               (square dark swept large up) (square dark swept large down)
               (square dark square small up) (square dark square small down)
               (square dark square large up) (square dark square large down)
               (square light swept small up) (square light swept small down)
               (square light swept large up) (square light swept large down)
               (square light square small up) (square light square small down)
               (square light square large up) (square light square large down)
             )))
         response properties)
        (dolist (stim stimlist)
           (setq response
                 (if (or (and (eq (first stim) 'down) (eq (second stim) 'dark))
                         (and (eq (first stim) 'down) (eq (third stim) 'swept))
                         (and (eq (second stim) 'dark) (eq (third stim) 'swept)))
                     'USA
                     'AUS))
           (setq properties
                 (list 'nose (first stim) 'shading (second stim) 'wings (third stim)
                       'tail (fourth stim) 'wheels (fifth stim)))
           (classify-one properties response)
)))


;;; ......

;;; This is a temporary replacement for RHS function otherwise given below

(defun response-given (class)
   (if (not (eql class true-class)) (incf total-errors)))


;;; ......

;;; check whether feedback is available

;(defun feedback-is-available () (< stimulus-number 41))
(defun feedback-is-available () T) ; ***************** temporarily **************
```

```
;;; get the correct classification

(defun correct-classification () true-class)

(clear-all)
(sgp
   :era t
   :pl t
   :ol t
   :egs 0.8
   :ut -1.5
   :rt 0.0
)

(chunk-type classify nose shading wings tail wheels class)
(chunk-type the-stimulus nose shading wings tail wheels class)
(chunk-type make-guess oldgoal class)
(chunk-type konstant)

(add-dm
   (maingoal isa classify)
   (the-stimulus isa the-stimulus)
   (AUS isa konstant)
   (USA isa konstant)
   (square isa konstant)
   (down isa konstant)
   (light isa konstant)
   (dark isa konstant)
   (swept isa konstant)
   (large isa konstant)
   (small isa konstant)
   (up isa konstant)
   (dum isa konstant)
   (unencoded isa konstant)
)


;;; Goal Structure

;;; main goal: classify
;;; subgoal:   make-guess

;;; Start with a totally unencoded description

(p all-unencoded            ;; **** currently not used ***
   =goal>
      isa      classify
      nose     nil
      shading  nil
      wings    nil
      tail     nil
      wheels   nil
==>
   =goal>
      nose     unencoded
      shading  unencoded
      wings    unencoded
      tail     unencoded
      wheels   unencoded
)
```

```
;;; Given a new stimulus, encode the attributes independently
;;; The idea is that only some of these encoding rules will fire.
;;; Hopefully, the model will learn which attributes are relevant, and encode only
those.

(p encode-nose
   =goal>
      isa      classify
      nose     unencoded
      class    nil
   =the-stimulus>
      isa      the-stimulus
      nose     =nose
==>
   =goal>
      nose     =nose
)

(p encode-shading
   =goal>
      isa      classify
      shading  unencoded
      class    nil
   =the-stimulus>
      isa      the-stimulus
      shading  =shading
==>
   =goal>
      shading  =shading
)

(p encode-wings
   =goal>
      isa      classify
      wings    unencoded
      class    nil
   =the-stimulus>
      isa      the-stimulus
      wings    =wings
==>
   =goal>
      wings    =wings
)

(p encode-tail
   =goal>
      isa      classify
      tail     unencoded
      class    nil
   =the-stimulus>
      isa      the-stimulus
      tail     =tail
==>
   =goal>
      tail     =tail
)
```

```
(p encode-wheels
   =goal>
      isa      classify
      wheels   unencoded
      class    nil
   =the-stimulus>
      isa      the-stimulus
      wheels   =wheels
==>
   =goal>
      wheels   =wheels
)


;;; During the test phase, there's no feedback, so just pop neither success nor fail

(p no-feedback
   =goal>
      isa      classify
      class    =class
   !eval!      (not (feedback-is-available))
==>
   !output!    (*** Predict =class ***)
   !eval!      (response-given =class)
   !pop!
)


;;; During the training phase ...

;;; ... if correct then pop with success

(p classify-correct
   =goal>
      isa      classify
      class    =class
   !eval!      (feedback-is-available)
   !eval!      (eq =class (correct-classification))
==>
   !output!    (*** Predict =class ***)
   !eval!      (response-given =class)
   !pop!
)


;;; ... if wrong then pop with failure

(p classify-wrong
   =goal>
      isa      classify
      class    =class
   !eval!      (feedback-is-available)
   !eval!      (not (eq =class (correct-classification)))
==>
   !output!    (*** Predict =class ***)
   !eval!      (response-given =class)
   !pop!
)
```

```
;;; Guess the classification

(p make-guess
   =goal>
      isa      classify
      class    nil
==>
   =guess>
      isa      make-guess
      oldgoal  =goal
   !focus-on!  =guess
)


(p guess-USA
   =goal>
      isa      make-guess
      class    nil
==>
   =goal>
      class    USA
)

(p guess-AUS
   =goal>
      isa      make-guess
      class    nil
==>
   =goal>
      class    AUS
)


;;; During the test phase, there's no feedback, so just pop

(p guess-no-feedback
   =goal>
      isa      make-guess
      class    =class
   !eval!      (not (feedback-is-available))
==>
   !output!    (*** Predict =class ***)
   !eval!      (response-given =class)
   !pop!
)
```

```
;;; During the training phase, learn from the guess

(p learn-from-guess
   =goal>
      isa        make-guess
      oldgoal    =oldgoal
      class      =class
   !eval!        (feedback-is-available)
   =oldgoal>
      isa        classify
      nose       =nose
      shading    =shading
      wings      =wings
      tail       =tail
      wheels     =wheels
      class      nil
==>
   !output!      (*** Predict =class ***)
   !eval!        (response-given =class)
=end>
      isa        classify
      nose       =nose
      shading    =shading
      wings      =wings
      tail       =tail
      wheels     =wheels
      class      (!eval! (correct-classification))
   =dependency>
      isa        dependency
      goal       =oldgoal
      modified   =end
      dont-cares unencoded
      specifics  (square down light dark swept large small up)
   !focus-on!    =dependency
)

(P pop-dependency
   =goal>
      ISA         dependency
==>
   !pop!
)
(spp no-feedback  :success t :failure t)
(spp classify-wrong  :failure t :eventual-efforts 10000 :eventual-successes 10000)
(spp make-guess  :success t :failure t :eventual-efforts 20000 :eventual-failures 8000
       :eventual-successes 12000)
(spp encode-nose  :eventual-successes 112 :eventual-failures 48 :eventual-efforts 160
       :effort 0.25)
(spp encode-shading  :eventual-successes 112 :eventual-failures 48 :eventual-efforts 160
       :effort 0.25)
(spp encode-wings  :eventual-successes 112 :eventual-failures 48 :eventual-efforts 160
       :effort 0.25)
(spp encode-tail  :eventual-successes 112 :eventual-failures 48 :eventual-efforts 160
       :effort 0.25)
(spp encode-wheels  :eventual-successes 112 :eventual-failures 48 :eventual-efforts 160
       :effort 0.25)
(spp guess-USA  :eventual-successes 10000 :eventual-failures 10000 :eventual-efforts 20000)
(spp guess-AUS  :eventual-successes 10000 :eventual-failures 10000 :eventual-efforts 20000)
```